



## PATENT ABSTRACTS OF JAPAN

(11) Publication number: **07170390 A**(43) Date of publication of application: **04.07.95**

(51) Int. Cl. **H04N 1/387**  
**G06T 11/80**  
**G06T 11/60**  
**H04N 1/41**  
**H04N 7/24**

(21) Application number: **06216828**(22) Date of filing: **12.09.94**(30) Priority: **20.09.93 US 93 124250**(71) Applicant: **RICOH CO LTD**

(72) Inventor: **ROBAATO EFU MIRAA**  
**MAATEIN PII BOORITSUKU**  
**SUTEIIBUN EMU BURONSUTAIN**

(54) **VIRTUAL EDITING METHOD AND DEVICE FOR COMPRESSED IMAGE**

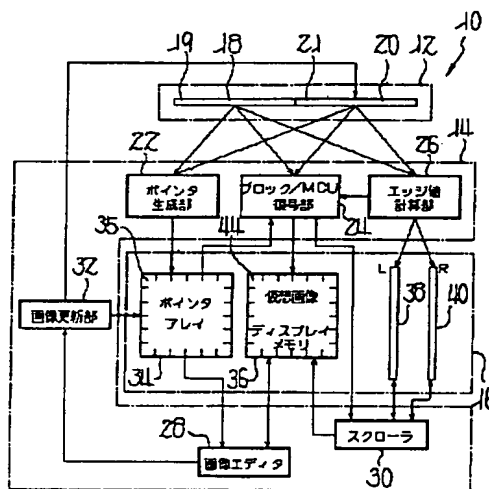
which is related to the block of the image 44 may be centralized.

## (57) Abstract:

COPYRIGHT: (C)1995,JPO

**PURPOSE:** To enable the image operation on an image picture element level as the image is compressed by editing the part of a compressed image without extending the whole image.

**CONSTITUTION:** A pointer generation part 22 performs a scanning from an old edition to the latest edition within an area 20, and when plural editions for a single block exist in the area 20, only the latest minimum encoding unit MCU for the block is shown by the pointer within a pointer array 34. Normally, an image edition is related to the slight part of the whole image and this part is operated as a complete image in an image editor 28 after a virtual image is decoded. A virtual image 44 is the one selected for edition and a block decoding part 24 decodes only 1/6 block of the image 44. This block is stored in a display memory 36, and the image 44 is processed from the memory 36 and is displayed or the image is preserved. As a result, because an image processing part 14 can shorten the time required for all the extensions of the whole image and an edge table or a DC value array is possible to be utilized, the only MCU



(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平7-170390

(43) 公開日 平成7年(1995)7月4日

(51) Int.Cl.<sup>8</sup>

H 0 4 N 1/387

G 0 6 T 11/80

11/60

識別記号

庁内整理番号

F I

技術表示箇所

9071-5L

G 0 6 F 15/ 62

3 2 0 A

9071-5L

3 2 5 A

審査請求 未請求 請求項の数18 O L (全 18 頁) 最終頁に続く

(21) 出願番号

特願平6-216828

(22) 出願日

平成6年(1994)9月12日

(31) 優先権主張番号

0 8 / 1 2 4 , 2 5 0

(32) 優先日

1993年9月20日

(33) 優先権主張国

米国 (U S)

(71) 出願人 000006747

株式会社リコー

東京都大田区中馬込1丁目3番6号

(72) 発明者

ロバート エフ ミラー

アメリカ合衆国 カリフォルニア州 メン

ロー パークスイート 115 サンド ヒ

ル ロード 2882 リコー コーポレーシ

ョン カリフォルニア リサーチ センタ

内

(74) 代理人

弁理士 柏木 明 (外1名)

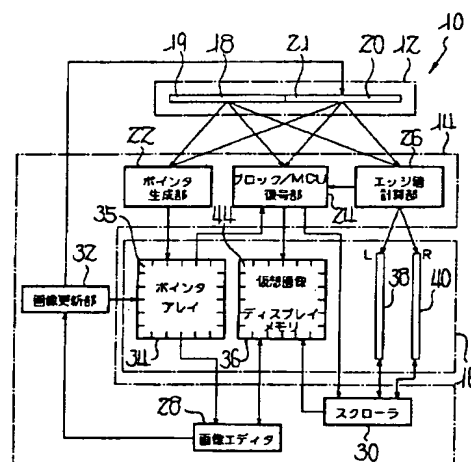
最終頁に続く

(54) 【発明の名称】 圧縮画像の仮想編集方法及びその装置

(57) 【要約】

【目的】 圧縮された形状のままの画像の画素レベルでの操作を可能にすること。

【構成】 典型的な画像圧縮方式を用いる画像圧縮システムにおいて、ポインタ・アレイ34が圧縮画像ファイル内の多くのMCUの各々を指し示すように提供される。画像の全てのブロックからブロックの部分集合が仮想画像として選択される。仮想画像は編集され、各々の編集されたブロックは編集ブロック内に圧縮される。編集済みブロックは編集済みMCU内に圧縮され編集ブロック領域内に配置されて、原本のMCUへのポインタは新規のMCUを指すように変更される。この方法によって、ポインタ・アレイ34は取消動作を実行するように変更することが可能である。



## 【特許請求の範囲】

【請求項1】 編集用に圧縮画像の一部を取り扱うための方法であって、前記画像は各々がデジタルデータによって表現されたブロックに分割されており、また、各々のブロックでの前記デジタルデータはMCU（最小符号化単位）に圧縮されており、複数の前記MCUが圧縮された画像データファイルを構成するようになしてあり、ポインタ・アレイを生成することであって、このポインタ・アレイ内の各々のポインタは前記圧縮された画像データファイルの内部のMCUを指し、圧縮されたMCUはMCUヘッダとMCU本体を含み、前記MCUヘッダはさらに少なくとも1つの画像ブロック識別子と前記MCU内のビット数を表わす長さフィールドを含み、前記MCU本体は少なくとも前記MCU本体の各画像ブロックの品質についての1つの相対値と複数の絶対値を含むような段階と、

前記画像からブロックを選択して仮想画像を定義する段階と、

参照テーブルを生成することであって、この参照テーブルは少なくともブロックの前記相対値をそのブロックの絶対値へ変換するために使用する1つの値を含む段階と、

前記仮想画像の前記ブロックに対応する1組のMCUを展開する段階と、

前記仮想画像を編集して少なくとも1つの編集済みMCUを形成する段階と、

前記編集済みMCUを編集済みMCU中に圧縮する段階と、

前記編集済みMCUを編集済みブロック・メモリ内に保存する段階と、

前記ポインタ・アレイ内のポインタを変更し、このポインタを前記編集済みブロックに対応させ、前記編集済みブロック・メモリ内の前記編集済みMCUを指すように変更する段階と、

を含み、

この変更段階で変更したように前記ポインタ・アレイの前記ポインタによって指し示される前記MCUが全体として編集済み画像を定義することになるようにしたことを特徴とする圧縮画像の仮想編集方法。

【請求項2】 編集用に圧縮画像の一部を取り扱うための方法であって、前記画像は各々がデジタルデータによって表現されたブロックに分割されており、また、少なくとも1つのブロックについての前記デジタルデータはMCU（最小符号化単位）で表現されており、複数の前記MCUが圧縮された場合に圧縮画像データファイルを構成するようになしてあり、ブロックは少なくとも前記ブロックの品質についての差分符号化した値を1つ含み、

ポインタ・アレイを生成し、このポインタ・アレイ内の各々のポインタが前記圧縮画像データファイルの内部の

MCUを指す段階と、

前記画像からブロックを選択して仮想画像を定義する段階と、

選択した前記ブロックの各々について差分符号化されている前記品質についての絶対値を求める段階と、

前記MCUの組のデータと前記絶対値を用いて、前記仮想画像の選択した前記ブロックに対応する1組のMCUを展開する段階と、

前記仮想画像を編集して少なくとも1つの編集済みMCUを形成する段階と、

前記編集済みブロックを編集済みMCU中へ圧縮する段階と、

前記編集済みMCUを編集済みブロック・メモリ内に保存する段階と、

前記ポインタ・アレイ内のポインタを変更し、このポインタを前記編集済みブロックに対応させ、前記編集済みブロック・メモリ内の前記編集済みMCUを指すように変更する段階と、

を含み、

この変更段階で変更したように前記ポインタ・アレイの前記ポインタによって指し示される前記MCUが全体として編集済み画像を定義することになるようにしたことを特徴とする圧縮画像の仮想編集方法。

【請求項3】 選択したブロックに対応するオフセットのエントリーを指し示すことによってオフセット・エントリーのテーブルを参照する段階と、

指し示されたオフセット・エントリーを差分符号化した値に加算して、選択した前記ブロックの前記品質についての前記絶対値を得る段階と、

を前記絶対値を求める段階が含むことを特徴とする請求項2記載の圧縮画像の仮想編集方法。

【請求項4】 選択したブロックに対応する絶対値エントリーを指し示すことによって絶対値エントリーのテーブルを参照する段階と、

選択した前記ブロックの前記品質についての前記絶対値として指し示された前記絶対値エントリーを使用する段階と、

を前記絶対値を求める段階が含むことを特徴とする請求項2記載の圧縮画像の仮想編集方法。

【請求項5】 定数オフセットを差分符号化した値に加算して前記選択したブロックの前記品質についての前記絶対値に達する段階を、前記絶対値を求める段階が含むことを特徴とする請求項2記載の圧縮画像の仮想編集方法。

【請求項6】 前記編集段階の前に編集しようとする前記ブロック全部について前記絶対値を求める段階を実行し、前記編集段階の前に絶対値で前記差分符号化した値を置換する段階を、さらに含むことを特徴とする請求項2記載の圧縮画像の仮想編集方法。

【請求項7】 編集後に前記絶対値を差分値で置換して

JPEG互換画像ファイルを得る段階を、さらに含むことを特徴とする請求項6記載の圧縮画像の仮想編集方法。

【請求項8】 ブロックの前記差分符号化した品質は前記ブロックが包括する画素の色相成分についての画素色相値のコサイン変換から得られたDC値であることを特徴とする請求項2記載の圧縮画像の仮想編集方法。

【請求項9】 編集用に圧縮画像の一部を取り扱うための方法であって、前記画像はブロックに分割されており、各々のブロックは、少なくとも1つの相対値を含むデジタルデータによって表現され、ブロックについての絶対値が他のブロックについての相対値を参照して求められるようになしてあり、

前記画像からブロックを選択して前記圧縮画像の前記部分に対応する仮想画像を定義する段階と、

選択自在な複数のブロックの各々について絶対値を求める段階と、

前記絶対値を保持するためのオフセット・テーブルを生成する段階と、

前記オフセット・テーブルを用いて展開し前記仮想画像に追加したブロックについての絶対値を求める段階と、を含むことを特徴とする圧縮画像の仮想編集方法。

【請求項10】 複数のMCU（最小符号化単位）を含む圧縮画像データファイルを記憶するための入力メモリであって、前記圧縮画像データファイルの各々のMCUはアドレス自在かつ独立して展開自在であり、MCUは前記MCU内のブロックが包括する1組の画素の少なくとも1つのブロックの少なくとも1つの色相成分について画素色相値を符号化し、また、ブロックは圧縮時に前記ブロックが包括する少なくとも1つの画素の品質についての少なくとも1つの相対値を含むとともに、圧縮出力データファイルを記憶するための出力メモリと、

初期化時に各々が前記入力メモリ又は前記出力メモリのどちらかのMCUを指し示す複数のポインタを含むポインタ・アレイと、

ブロックの前記品質について前記少なくとも1つの相対値から絶対値への変換の際に用いる少なくとも1つの値を含む参照テーブルと、

前記ポインタ・アレイと前記入力メモリに結合してあり、前記ポインタ・アレイ内のポインタによって示された位置で前記入力メモリから圧縮MCUを読み出すための選択手段と、

この選択手段と前記参照テーブルに結合してあり、圧縮MCUの内容と前記参照テーブルから読み出した少なくとも1つの値に基づいて画素色相値の少なくとも1つのブロックへ少なくとも1つのMCUを展開するための展開手段と、

この展開手段に結合してあり前記展開手段によって提供されたMCUの画素を編集するための画像編集手段であ

って、この画像編集手段は編集したMCUを出力するための手段と、

を含むことを特徴とする圧縮画像の仮想編集装置。

【請求項11】 前記参照テーブルは単一のエントリーを含み、このエントリーは前記展開手段がブロックの相対値に加算して前記ブロックの品質についての絶対値を求めるための定数値を保持するようになしてあることを特徴とする請求項10記載の圧縮画像の仮想編集装置。

【請求項12】 前記参照テーブルは前記画像データファイル内にブロック毎に1つのエントリーを含み、前記展開手段は前記エントリーに記憶してある値をブロックの相対値に加算して前記ブロックの品質についての絶対値を求めるようになしてあることを特徴とする請求項10記載の圧縮画像の仮想編集装置。

【請求項13】 前記参照テーブルは前記画像データファイル内のブロック毎に1つのエントリーを含み、前記展開手段は前記エントリーに記憶してある値を前記ブロックの品質についての絶対値として用いるようになしてあることを特徴とする請求項10記載の圧縮画像の仮想編集装置。

【請求項14】 前記参照テーブルは生成された後前記入力メモリ内に記憶され、前記参照テーブルの各々のエントリーは前記入力メモリ内のMCUのブロックに関連することを特徴とする請求項10記載の圧縮画像の仮想編集装置。

【請求項15】 前記参照テーブルの前記エントリーは前記入力メモリ内の各々のブロックの前記相対値を置き換えることを特徴とする請求項14記載の圧縮画像の仮想編集装置。

【請求項16】 完全な画像がM行N桁のブロックに分割され、各々のブロック内の画像がDCオフセット値を含む1組のデジタル値で表現され、前記DCオフセット値は前記ブロックについてのDC値と前記ブロックの左側のブロックについてのDC値の間の差を表わすようになしてあり、Y行X桁のブロックを含む仮想画像がディスプレイ上に表示され、前記ディスプレイ上の仮想画像は最上部の行のブロックと、底部の行のブロックと、左側の桁のブロックと、右側の桁のブロックで定義され、YはMより小さく、XはNより小さく、M行からY行が選択されN桁からX桁が選択されるようになしてある画像編集システムにおいて、

M個のエントリーを含み、各々のエントリーが前記仮想画像の前記左の桁の左側に隣接する桁にあるブロックのDC値に対応するように左側エッジ・テーブルを生成することと、

M個のエントリーを含み、各々のエントリーが前記仮想画像の前記右の桁の右側に隣接する桁にあるブロックのDC値に対応するように右側エッジ・テーブルを生成することと、

前記仮想画像が前記完全な画像に対して左に移動すると

き前記左側エッジ・テーブルを使用して前記仮想画像に追加するブロックについてのDC値を計算すること、前記仮想画像が前記完全な画像に対して右に移動するとき前記右側のエッジ・テーブルを使用して前記仮想画像に追加するブロックについてのDC値を計算すること、

を含むことを特徴とする圧縮画像の仮想編集方法。

【請求項17】 完全な画像がM行N桁のブロックに分割され、各々のブロック内に含まれる前記画像の一部がDCオフセット値を含む1組のデジタル値として表現され、前記DCオフセット値は前記ブロックについてのDC値と前記ブロックの左側のブロックについてのDC値の間の差分を表わすようになっており、Y行の前記ブロックを含む仮想画像がディスプレイ上に表示され、前記ディスプレイ上の前記仮想画像は左側の桁のブロックを有するような画像編集システムにおいて、M個のエントリーを含み、各々のエントリーが前記仮想画像の左側の桁の左側に隣接する桁のブロックのDC値に対応するように左側エッジ・テーブルを生成すること、

前記左側エッジ・テーブルを使用して前記完全な画像に対して前記仮想画像を左側に移動するとき前記仮想画像に追加されるブロックについてのDC値を計算すること、

を含むことを特徴とする圧縮画像の仮想編集方法。

【請求項18】 完全な画像が画素のパターンを含み、各々の画素が画像ファイル内のデジタル値で表現されており、また、前記画像ファイルはMCU（最小符号化単位）を含み、各々のMCUは画素のグループ内の各々の画素についてのデジタル値を表わすような編集装置であって、

画像ファイルのMCUを圧縮MCUに圧縮するための圧縮部と、

前記圧縮MCUを展開MCUに展開するための展開部と、

前記画像ファイルからのMCUを保持するための原本ブロック・メモリと、

前記展開MCU内のデータを操作するためのエディタと、

このエディタで編集したMCUを再圧縮する前記圧縮部から得られた編集済みMCUを保持するための編集済みブロック・メモリと、

前記MCUを前記原本ブロック・メモリ内に割り当てるために前記再圧縮した編集済みMCUを前記画像に割り当てるためのポインタのアレイと、

を含み、

前記エディタには前記画像ファイル全体が展開されたように選択した展開MCUが提供されるようにしたことを特徴とする圧縮画像の仮想編集装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明は、圧縮画像の編集分野に関し、より特定すれば、画像全体を展開することなく圧縮画像の部分編集のための圧縮画像の仮想編集方法及びその装置に関する。

【0002】

【従来の技術】 カメラ及びスキャナにとって、現在のカラー画像は、入力し又は走査した画像を表現するデータを含む画像ファイルを保存するために、大量のメモリを必要とする。カラー複写機において、8.5インチ×11インチ（216mm×297mm）の用紙、解像度400ドット/インチ（15.75ドット/mm）、各ドットの色を表現するために3バイト（24ビット）を用いるような典型的な画像では、44.88メガバイトのメモリを占有することになる。メモリの容量に加え、メモリはリアルタイムの用途、例えば、毎分15枚の速度が一般的な要求とされるカラー複写機などのための広い帯域幅を有する必要がある。これは、帯域幅又はデータ速度でほぼ90メガビット/秒に相当することを意味する。

【0003】 これら双方の要求が画像システムのコストを押し上げることになる。圧縮技術は、当面の帯域幅と保存容量の問題の両方を解決するために用いられてきた。圧縮によると、画像ファイルは少ないメモリに保存可能であり、画像を表現するのに使用されているビット数が少ないため、帯域幅の制限されている何らかのチャンネル間で画像を高速に転送することが可能である。圧縮は元の画像データをコンプレッサ回路に通すことで行なわれ、コンプレッサ回路が元のデータに含まれるパターンを統合又は分析して、圧縮画像ファイルを生成するが、対応する展開なしには元の画像が容易に識別し得ない。

【0004】 画像に編集、伸縮、回転又はその他の処理を加える場合、何らかの又は全ての画素への無作為なアクセスが要求される。処理が完了すれば、編集結果を画像ファイルに保存する必要がある。処理している画像ファイルが圧縮されている場合、従来の無作為な画素へのアクセスを得るための手段は、画像をフレーム記憶装置内に展開し、処理を実行し、しかる後、画像を新しい画像ファイルに再圧縮する。これに伴う問題は、展開によりメモリが保存されないことで、フレーム記憶装置にメモリが必要とされることである。画像全体の圧縮と展開に時間がかかることから帯域幅も低下する。画像システムにおいて最も高価な表示装置でさえも完全な解像度で画像全体を表示し得ないことから、このさらなる処理時間とメモリはしばしば無駄となる。

【0005】 画像データを圧縮するための圧縮法は周知である。こうした圧縮規格の1つがJPEG（Joint Photographic Experts Group）である。その他の規格としては、JBIG、ファクシミリG3又はG4規格、及び、GIFが含まれる。JPEGでは、1つの例

を示すと、画像は画像の要素又は画素の2次元アレイによって表現されている。画像がグレースケールの場合、各々の画素は濃度値により表現され、また、画像がカラーの場合、画素は幾つかの成分画像（例えば、RGB色分離規格に従って分割した画像では、赤、緑、青の成分など）に分割され、各々の成分画像の各々の画素が成分の色の値で表現される。

【0006】画像データ（本明細書では、画像からのデータを含めて画像ファイルと称する）を圧縮画像ファイルに圧縮する以前に、画像は成分に分解され（画像が白黒でない場合。白黒画像の場合には1つの成分だけしかない）、しかる後、各々の成分は8×8画素又は64画素からなる正方形の画像領域を各々が包括するようなブロックに分割される。各ブロック内で64画素はディスクリット・コサイン変換（DCT変換）を用いて直流（DC）値と63の交流（AC）値からなる64の周波数振幅に変換される。これらの値が圧縮画像データに符号化される。

【0007】高い圧縮比を実現するには、あるブロックのDC値とその直前のブロックのDC値の差をそのブロックの絶対DC値の代わりに使用する。大半の画像では絶対値より差分値で表わす方が少ないビット数で足りるので、画像データ全体として高い圧縮比を得ることができる。このような理由から、JPEG以外の多くの画像圧縮方式も同様の圧縮技術を使用し差分値を用いている。

【0008】画像の圧縮率を向上させるために色のサブサンプリングを用いることが多い。画像がYUV色空間（Y, Cr, Cb）の3つの成分に分解されるとき、Y成分は輝度（Y）を表わし、U又はCr成分は赤-緑の色相成分、また、V又はCb成分は青-緑の色相成分を表す。視覚的に輝度成分は最も重要な成分で、画像に所望しないアーチファクトを導入することなく他の2つの成分の解像度を減少させられることが多い。この輝度解像度に対する色相解像度の減少はサブサンプリングとして周知となっている。各々の色成分（UとV）の解像度を半分に減少させることを2:1:1サブサンプリングと称するが、これは、2つの輝度（Y）画素色相値の各々に対して一对の（U, V）画素色相値だけが提供されるためである。サブサンプリングなし（1:1:1）であれば、1画素の領域は3つの画素（1L, 1U, 1V）全体として記述されるが、2:1:1サブサンプリングを行なうと、2画素の領域を4つの画素（2L, 1U, 1V）で記述することができ、画像データファイルの大きさを2/3に減少させられる。4:1:1サブサンプリングを行なうと、4画素の領域が12画素ではなく6画素で記述でき、画像データファイルの大きさを1/2に減少できる。

【0009】サブサンプリングを用いる場合、異なる色成分のブロックが異なる大きさの領域に跨るので、成分

のブロックを一緒に保持するためには、符号化と復号化の際に小さな領域について全てのブロックを一緒に保持する。例えば、2:1:1サブサンプリングを行なうとき、4ブロック、即ち、Uブロックと、Vブロックと、2つのYブロックが8画素×16画素の領域を包括するのに必要であるから、これら4ブロックを一緒に保持する。4:1:1サブサンプリングの場合、16×16画素の領域は6ブロック（4Y, 1U, 1V）で包括されるので、これらを一緒に保持する。この圧縮展開自在なデータの最小単位はMCU（最小符号化単位）として公知である。JPEG規格によれば、一度に1つのMCUだけ圧縮画像ファイルからデータを抽出できる。

【0010】圧縮処理は、ブロックに含まれる64画素の色相値の各々を1つの差分DC値と63個のAC値に変換することと、複数ブロックMCUとしてブロックをMCUにまとめることと、しかる後、エントロピー符号化によって各々のMCUを記述するのに必要なビット数を減少させることと、から構成される。圧縮画像を構成する圧縮MCUは、例えば、展開時の画像内でのMCUの位置、MCU内に含まれるビット数を表わす長さデータ、及び、同様な情報などの識別情報でタグを付け、MCUを圧縮画像ファイルとして順次保存する。長さデータが必要なのはMCUが可変長であり、長さが圧縮しようとするデータと用いる圧縮方法の関数をなすためである。エントロピー符号化処理は、一般に符号化を指し、ブロック化と、コサイン変換と、画像データ符号化の処理を指す。

【0011】以上から明らかなように、画像のブロックについての差分DC値は画像データのエントロピー復号の段階の後圧縮画像から抽出することができ、このほうがデータの逆コサイン変換のさらなる段階を必要とする画像の完全な展開より労力が少ない。あるブロックのDC値を得るために逆変換しなくてもよいことから、DC値を得るのに必要な労力は展開したブロックを得る場合のそれに比べて大幅に少ない。これは、コサイン変換と逆変換が非常に計算資源を多用する演算であるためである。

【0012】エントロピー復号を圧縮MCUに適用すると、結果は各々のブロックについての差分DC値と63個のAC値となる。次に、差分DC値に注目しているブロックの左側のブロックの絶対DC値を加算して絶対DC値へ変換する。当然のことながら、このDC値再生処理を機能させるためには、注目しているブロックの左側のブロックが全て処理される必要がある。次に、ブロックを逆コサイン変換に通して圧縮されていない画素色相値データを得る。

【0013】DC値などのブロック間で緩やかに変化する値の差分符号化は圧縮を増加させるが、画像の一部の展開の困難さも増大させる。これは、部分的画像の幾つかのMCUの展開が部分的画像の一部をなさないMCU

の参照を必要とするためである。

【0014】部分的画像が編集され再圧縮される場合には、さらなる困難が部分的展開で発生する。圧縮度は画像データの順序に依存しているため、MCUの変化を表す画像部分として大きさが変化することになる。その結果、単純に部分的画像からのMCUを編集した部分的画像からのMCUで置換することは、画像全体が展開、編集、再圧縮されない限り非実用的である。

【0015】

【発明の目的】上記のことから、圧縮画像を操作し編集するための方法の改良が必要とされていることは明白である。

【0016】

【発明の概要】本発明は、圧縮された形状のままの画像の画素レベルでの操作を可能にする。典型的な画像圧縮方式を使用する画像圧縮システムにおいては、ポインタで圧縮画像ファイル中のMCUを示すポインタ・アレイを用いており、幾つかの発明においては、DC値テーブルを設けてあるブロックの絶対DC値が他の何らかのブロックにあるデータを参照する必要なしに計算できるようにしている。画像のブロック全体からブロックの部分集合が仮想画像として選択される。幾つかの発明において、仮想画像は1つの仮想画像より大きい画像領域を包括するように移動し、場合によっては、画像領域全体を順次包括する。

【0017】仮想画像を展開／復号して編集し、編集した各々のMCU（即ち、これの1つ又はそれ以上のブロック）を編集したMCUに圧縮し、編集したMCUを指すように本来のMCUへのポインタを変更して、編集した画像データ領域に保存する。この方法では、ポインタ・アレイは取消操作を実行するように変更することが可能で、対話型画像編集アプリケーションにおいて有用である。当然のことながら、対話型画像編集アプリケーションの代わりに、独立した出力画像ファイルを作成する逐次処理によって入力画像ファイル全体を編集することも可能である。

【0018】本発明は、圧縮MCUから全てのMCUが復号されないようなMCU内の差分符号化した値を取り扱うための手段を提供する。値を保持するためにエッジ・テーブルを用意し、ここで各々の値は仮想画像の辺縁上のブロックについての差分値と結合された場合、仮想画像の辺縁より遠方のブロックを参照することなくそのブロックについての絶対値を提供する。エッジ・テーブル内のエントリは、ブロックを完全に展開せずに圧縮されたままのMCUから決定される。これは、逆コサイン変換がエッジ・テーブルの作成に必要なためである。さらなる記憶空間の必要性より処理速度が重要な場合に差分値から絶対値を計算する速度を向上させるため、1つ以上のエッジ・テーブルを設けることができる。絶対値計算は平均してエッジ・テーブルが多いほど

高速であるが、これは、計算速度がエッジ・テーブルを有する最も近い辺縁までのブロックからの距離に比例するためである。

【0019】例えば、各々のブロックについてのDC値などの差分符号化した値の記憶空間に対して処理速度が特に優先する場合、画像を含む全てのブロックについてのDC値を抽出してDC値アレイに保存する。

【0020】処理速度と記憶空間が両方とも優先する場合、さらに別の発明において差分DC値は一定のDC値に対する差分としている。本発明の利益は、JPEG互換ではないが、各々のMCUが独立して展開自在であり、エッジ・テーブル又はDC値アレイが不要な点である。本発明で用いる画像ファイルとJPEG互換画像ファイルの間の画像の変換は計算が簡単で、無損失、反転自在な演算である。

【0021】本発明は、また、全てのMCUを並べ替え原本の画像データ領域からのMCUを編集した画像データ領域からのMCUで置換するために圧縮された編集画像の再インデックスを取るための手段も提供する。本発明で解説する好適実施例は、JPEG規格に従って圧縮された画像データで動作するが、本発明は、ブロックが可変容量MCUに圧縮されるような、又は、データが差分値で符号化されることによりMCUを完全に復号するためには別のMCUを参照しなければならないようなあらゆる画像圧縮方式について応用が可能である。

【0022】本発明による画像エディタにおいて、圧縮画像は主メモリ領域内に読み込まれ、また、編集するために選択したブロックが主メモリ領域から展開されてエディタに渡され、編集され、圧縮回路によって再圧縮される。再圧縮され、編集されたブロックは編集ブロックメモリ内に配置される。ポインタ・アレイを用いて、編集されたブロックメモリ又は主メモリ領域内の何れかのブロックを示し、アレイ内のポインタが編集後の画像の各々のブロックを指し示すようにする。当然のことながら、全てのMCUを編集し逐次的に置換する処理の場合などで、選択したMCUは画像のあらゆるMCUを含むことがある。

【0023】編集の方法は、回転、変換、伸縮、フィルタ処理、マスキング及び混合を含むが、これに制限されるものではない。マスキングは、各々の画素でマスキング値を評価するマスキング関数に従って画像を切り取る処理である。混合は、混合しようとする各々の画像について各々の画素で重み値を評価する混合関数に従って、画素毎に2つの画像を結合する処理である。

【0024】

【実施例】本発明の一実施例を図面に基づいて説明する。まず、図1は本発明による編集システム10の好適実施例を機能的に図示したものである。この編集システム10は、圧縮画像メモリ12、画像処理部14及び画像メモリ16を含む。前記圧縮画像メモリ12は、圧縮

画像19を含む原本の画像データ領域18と編集したMCU21を保持するための編集画像データ領域20を含む。画像処理部14は、ポインタ生成部22とブロック／MCU復号部24とエッジ値計算部26と画像エディタ28とスクローラ30と画像更新部32とを含む。画像処理部14のこれらの部材は、独立したハードウェアとして、或いは、処理装置により実行されるソフトウェアのサブルーチン群として、又は、両者の組合せとして実現することが可能である。

【0025】前記編集システム10は、例えば、利用者が画像の一部を観察しつつマウス及び／又はキーボードを用いて編集命令を発行する場合などの対話型アプリケーションとして有用であるが、編集システム10は画像と命令が利用者の介入なしに編集システム10へ供給されるような画像処理システムの構成部材としても有用である。

【0026】画像メモリ16は、ポインタ・アレイ34、ディスプレイ・メモリ36、左側エッジ・テーブル38及び右側エッジ・テーブル40を含む。前記圧縮画像メモリ12はランダム・アクセス・メモリ(RAM)、磁気ディスク又はその他の記憶媒体などのデジタルメモリである。

【0027】前記圧縮画像メモリ12に何らかの編集を行なうまで、編集画像データ領域20は空で編集されたMCUを全く含まない。この編集画像データ領域20に割り当てられるメモリ量は、実行すべき編集の量に基づく利用者又はアプリケーションの必要性和多重取消動作の必要性に従って変化し得る。

【0028】前記画像メモリ16の初期化についてこれから説明する。画像メモリ16は、編集作業の開始時又は圧縮画像メモリ12内に新規に圧縮画像が読み込まれた場合に初期化される。画像はブロックの2次元アレイよりなる。カラー画像において、画像は、第1に複数のカラー層に細分され、各々のカラー層は単一の画像として扱われる。以下の議論では、単一のカラー層画像についてのみ説明するが、説明する原理は直接的に複数のカラー層に拡張し得るものである。ブロックは通常8×8画素の方形を含み、非圧縮時には64の濃度値を提供しており、また、画像はこのようなN×M個のブロックのアレイにより構成される。各々のブロックが8×8画素であるため、8.5インチ×11インチでインチ当たり400画素の場合、Nは425、Mは550となり、画素当たりの成分が単一と仮定すると、全体として画像当たり233、750ブロックとなる。この例について続けると、圧縮画像19は圧縮データを含む233、750MCUを含むことになる。この例ではMCUは1ブロックとなる。

【0029】圧縮画像メモリ12内の画像が編集されて再インデックスが取られなかった場合、編集画像データ領域20は圧縮データのMCUを含み、領域20内の各

々のMCUで原本画像データ領域18内のMCUを置換する。また、ブロックが1回以上編集された場合、領域20は領域18のMCUを置換するための1つ以上のブロックを含むことになるが、実際に領域18内のブロックを置換するのは1つの編集されたMCUだけである。ポインタ生成部22は領域18を走査してポインタ・アレイ34内に各々のMCUの開始部分へのポインタを配置する。ブロックとポインタの1:1対応により、ポインタ・アレイ34はN×M又は233、750個のエントリーを含む。

【0030】図2及び図3は、ポインタ・アレイ34内のポインタがどのように構成されるかをよく図示している。第1のポインタ35は圧縮画像19の第1のMCUを示す。他のポインタも同様に、ブロックとそれらに対応するMCUに付随している。ポインタ・アレイ34は、図1に図示したように2次元アレイ内に論理的に配置して画像内のブロックの構成に対応させることができ、又は、図3に図示したように1次元アレイ内に構成することが可能である。各々のポインタは領域18内のMCUを示すか、又は、ポインタ(n+3)の場合のように領域20内のMCUを示す。図2及び図3のブロックの番号はMCU当たり1つのブロックを仮定したものであるが、この例は複数ブロックMCUに容易に拡張できる。

【0031】図1をさらに参照すると、領域18内の各々のMCUを走査してから、ポインタ生成部22は編集画像データ領域20を走査する。領域20内の各々のMCUは別のMCU内のデータを置換する画像データを含む。置換されたMCUは、MCUが原本のMCUの第1の編集物である場合には、領域18内のMCUを置換するか、又は、MCUが既に編集されている場合、領域20内のMCUを置換する。領域20内の各々のMCUは置換されるMCUへのポインタを含む。従って、ポインタ生成部22は領域20内の各々のMCUを単純に走査し、そのMCUに関連するブロックを認識してポインタ・アレイ34内のそのブロックへのポインタを更新する。ポインタ生成部22は古い編集から最新の編集へと領域20内を走査し、単一ブロックについて複数の編集が領域20内に存在する場合、そのブロックについて最新のMCUだけがポインタ・アレイ34内のポインタによって示される。

【0032】ポインタ・アレイ34が初期化されると、ブロック復号部24はアレイ34内のポインタを用いて領域18内の選択されたMCUを検索するが、別の実施例において、ブロック復号部24はポインタ・アレイ34の参照なしに領域18内のMCUを特定している。しかし、処理の重複を回避する上で、ポインタ・アレイ34を用いることが望ましい。ブロック復号部24は圧縮画像メモリ12内に保存された画像全体を復号するが、本発明の幾つかの利点は失われることになる。



【0033】通常、画像の編集は、全画像のわずかな部分にのみ関連する。この部分は、仮想画像が復号された後で画像エディタ28により完全な画像として操作される。図2は仮想画像44が全体画像42にどのように関連するかを図示している。図2において、全体画像42はN×Mブロックの広がりを持つが、仮想画像44は4×4ブロックの大きさである。

【0034】図1に戻って上記の例を続けると、仮想画像44は編集用に選択された画像であり、ブロック復号部24は仮想画像44の16ブロックだけを復号して得られたブロックをディスプレイ・メモリ36内に配置する。ディスプレイ・メモリ36から仮想画像44を処理し、表示させ、又はそのまま保存させることができる。明らかなように、画像処理部14は全体画像の全てを展開するのにかかる処理資源と時間を占有する必要がなく、エッジ・テーブル又はDC値アレイが本発明によって利用可能なため、仮想画像44内部のブロックに関連するMCUにだけ集中すればよいことになる。

【0035】初期化の間、ブロック復号部24はエッジ値計算部26と並行して動作している。他の実施例では、これらの動作は逐次的又は全く独立である。エッジ値計算部26は圧縮画像メモリ12内のブロックを走査し、差分であって基準ブロックへの参照を必要とする各々のブロック(MCU)の差分値だけを評価する。1つの規格であるJPEG規格では、各々のブロックはDC濃度についての差分値を含み、基準ブロックは復号されるブロックの左にあるブロックである。例えば、ブロック(n+2)についての絶対DC値を検索するには(図2参照)、差分値がブロック(n+2)から抽出され、抽出された値からブロック(n+1)についての絶対DC値が減算される。しかし、ブロック(n+1)は仮想画像44内に含まれないため展開していないので、エッジ値計算部26によりDC値だけがそのブロックから抽出する。全体画像に渡り仮想画像を移動させる際の遅延を回避するため、左右のエッジ・テーブル38, 40が用意されている。

【0036】図4は、エッジ・テーブル38, 40とディスプレイ・メモリ36内に保存されている仮想画像44の関連性を良く図示している。仮想画像44は当初にブロックA, B, C, Dにより仕切られている。左側エッジ・テーブル38は全体画像42の各行について1つのエントリーを含み、また、テーブルは仮想画像44の境界を越えて到達することが多い。各行についてのエントリーはその行の中のブロックについてと仮想画像44内の最も左の桁のブロックの左側までの絶対DC値を含む。例えば、左側エッジ・テーブル38のエントリーYはブロックDの左側に存在する全体画像中のブロックについてのDC絶対値である。同様に、右側エッジ・テーブル40は仮想画像44の最も右側の桁にあるブロックのDC絶対値についてのエントリーを含む。よって、エ

ッジ・テーブル38, 40は各々M個のエントリーを含むことになる。

【0037】仮想画像44が全体画像42に渡って水平方向にスクロールされる場合、更新領域100内の新しいブロックが展開され、ディスプレイ・メモリ36内に移動される。エッジ・テーブルを用いずにブロックを完全に展開するには、復号されるブロックの左側の各々のブロックについてのDC差分値が全て加算されなければならない。それに対して、右側エッジ・テーブル40を用いると、1回の加算又は減算だけでよいことになる。例えば、ブロックEが展開される場合、ブロックEでの絶対DC値はブロックEの差分DC値と右側エッジ・テーブル40内に記憶しているブロックBについてのDC絶対値から見つかる。各々の新しい桁が仮想画像44内に導入されるに従って、エッジ・テーブル40はその新しい桁についての絶対DC値で更新される。例えば、ブロックBについての絶対DC値を保持しているエントリーは1桁左へスクロールした後ではブロックEについての絶対DC値を保持することになる。他の圧縮方式においては、差分値が復号するブロックの上下のブロックを参照するような上部と底部のエッジ・テーブルを用いることがある。得られた仮想画像44と得られたエッジ・テーブルL'及びR'が図示してある。

【0038】2つのエッジ・テーブル38, 40は厳密には必要ではないが、大きな仮想画像の場合には、2つのエッジ・テーブル38, 40を用いるとスクロール中に必要な計算が減少する。例えば、右側エッジ・テーブル40が存在していない場合、ブロックEについての絶対DC値は左側エッジ・テーブル38から決定し、仮想画像44の最上行の各ブロックについての差分DC値を加算することになる。1つのエッジ・テーブルを用いる場合でも、エッジ・テーブルを有する仮想画像の辺縁と全体画像の辺縁の間のブロック数だけ必要な加算回数が減少する。エッジ値計算部26は仮想画像44内のブロックを含めたブロックについての絶対DC値を決定するので、これらの値はブロックを復号するためにこの値を使用するブロック復号部24に提供することができる。

【0039】仮想画像44をスクロールする場合、スクローラ30はエッジ・テーブル38, 40内のDC値を用いて仮想画像内でスクロールしようとするブロックについてのオフセットを提供する。例えば、全体画像の205桁目から204桁目へ仮想画像の左辺縁が移動するように仮想画像を移動した場合、新しい仮想画像内部の204桁目のブロックを復号してディスプレイ・メモリ36内に配置する必要がある。203回の減算を実行して204桁目のブロックの絶対DC値を求める代りに、204桁目の各々のブロックについての絶対DC値を左側エッジ・テーブル38の適切なエントリーから読み出すことができる。左側エッジ・テーブル38は203桁目の絶対DC値を保持するように更新される。これは、

204桁目の絶対DC値と204桁目のDC差分値から計算する。

【0040】同様に、仮想画像の右エッジが204桁目の場合、右側エッジ・テーブル40は205桁目の絶対DC値を含む。1桁右へスクロールする場合、スクローラ30はブロック復号部24に新しい仮想画像のブロックに対応するMCUを復号させる。新しいブロックは205桁目にあり、これらのブロックについての絶対DC値は右側エッジ・テーブル40内で簡単に見つかる。右側エッジ・テーブル40はこの後更新され、内部の各々のエントリが205桁目のブロックの差分DC値で変更され、206桁目のブロックについてのDC値が得られる。

【0041】仮想画像を上スクロールする場合、エッジ・テーブル38、40は変更せず、異なるエントリを使用する。エッジ・テーブル38、40は全体画像42の長さによって延在しているので、仮想画像はエッジ・テーブル38、40内のエントリをどれも変更することなく全体画像の上部から底部までスクロールすることができる。

【0042】画像エディタ28は、ディスプレイ・メモリ36の内容を観察しつつ画像を編集し得るような対話型装置とすることができ、又は、画像エディタ28は自動処理とすることもできる。1つの用途は、利用者がディスプレイ・メモリ36の内容を観察しながら又は観察せずに、利用者からの命令入力にตอบสนองする複写機内の画像の変更を行なうことである。編集内容を保存する場合、画像更新部32は編集MCUを再圧縮し、編集MCUを編集画像データ領域20内の利用可能な位置に保存し、そのブロックに関連するポインタ・アレイ34内のポインタを更新して、新規に保存した編集MCUを示させる。画像更新部32はまた新規保存したMCUがどのMCUを置換するかを表わす参照フィールドも編集MCUと一緒に保存する。この方法では、画像エディタ28が「取り消し」操作を起動した場合、画像更新部32は単にポインタを変更して、新規保存したMCUが参照するMCUを示すようにさせて、効率的に直前の編集を取り消すことができる。画像更新部32及び画像エディタ28は、また、1つ以上のブロックについて同時に操作する能力を有する。画像更新部32は各々の編集MCUについてのポインタ・アレイ34を更新するので、ポインタ・アレイ34は最新の状態にあり、新規の圧縮画像が圧縮画像メモリ12内に読み込まれるまで、圧縮画像メモリ12をポインタ生成部22で走査する必要がない。

【0043】図5はポインタ・アレイ34からのポインタ35を示す。前述の好適実施例において、ポインタは32ビットであり、編集フラグ50を保持するための最上位ビットと残り31ビットでMCUアドレスを構成する。「仮想編集」は、ツールを用いてディスプレイ・メ

モリ36内の画像データを変更することで実現している。ブロック内の画素を編集すると、そのブロックのポインタの編集フラグが設定される。編集は編集ブロックについての絶対DC値に影響を及ぼし得ることから、編集ブロックの右側のブロックのポインタ内の編集フラグも設定される。これは、右側のブロックでの差分DC値が左側のブロックでの絶対DC値を変化させるためである。編集作業が完了すると、ポインタ・アレイ34は設定されている編集フラグについて走査される。各々のフラグが立てられた領域では、圧縮画像メモリ12からブロックを復号し、フラグの立てられた領域のブロックの各列について左側に見えるブロックから開始して、各々の編集ブロックの左側のブロックでのDC値を決定する。編集したブロックが左側又は右側のエッジに到達した場合、DCエッジ・テーブルは新しい値で更新される。

【0044】図6に図示したように、新しい編集データは編集画像データ領域20内の原本の画像データ領域18に追加される。つまり、ポインタ・アレイ34は単一の線形メモリとして圧縮画像メモリ12をアドレスすることができる。編集MCUが作成されると、図6に示した編集ヘッダ・フィールドが初期化されて、編集MCUが領域20内に保存され、ポインタ・アレイ内のポインタが初期化されて新規保存した編集MCU内における編集ヘッダの始点を示す。

【0045】なお、図6中に示す各フィールドは、MRKR=JPEG APPn marker for scan and block id purposes (走査及びブロックID用のJPEG APPのマーカ)

30 HLEN=Length of block + in bytes to conform to JPEG standard (JPEG規格に適合させるためのブロック+2のバイト長)

EID =Edit id number for unique id signature when searching (検索時の独自IDシグネチャ用編集ID番号)

PBP =Previous bit position of previously edited same mcu (直前に編集した同一MCUの直前ビット位置)

40 NBP =Next bit position of subsequently edited same mcu (続けて編集した同一MCUの次のビット位置)

CBB =Number of bits comprising following encoded data (これ以下に符号化されているデータを含めたビット数)

CBD =Recompressed edited mcu image data (再圧縮された編集MCUの画像データ)

である。

【0046】MRKR及びHLENフィールドは、編集MCUがJPEG規格に適合するように提供されている。PBPフィールドはMCUの直前の版を指す。直前

の版は第1の編集MCUについて領域18内に存在するが、編集MCUがさらに編集された場合には、第2の編集MCUのPBPフィールドは第1の編集MCUを指すことになる。第1の編集MCUのNBPフィールドは第2の編集MCUの先を指すように設定されることになるので、リンクされたリストを構成する。CBBフィールドは後続する画像データのビット数を表わしているもので、所望するヘッダが見つかるまで1つのヘッダから別のヘッダへ飛び越し走査することにより画像ファイルが走査できる。CBDフィールドは画像データそれ自体を含む。さらにCBDフィールド内のデータはエントロピー符号化されているので、ビット数はMCU毎に異なる。これらのフィールドは、編集画像ファイルを新規画像ファイルに順次読み込むようにできるので、編集MCUが新規画像ファイルの原本データ領域内に含まれ、さらに、新規画像ファイルの編集領域が空になるように編集領域のMCUを画像ファイル内へ再配置する。

【0047】図7ないし図11は、編集システムの別の実施例を示す。この編集システム100では（特に、図10を参照）、入力ブロックを編集関数に従って処理しており、編集システム100はDC値アレイ又はDC定数オフセットのどちらかを使用してエッジ・テーブルの必要性を回避している。編集システム100は図1に図示した編集システム10の対話型編集及び取消機能を完全に支持する能力を有している。このシステム、編集システム10は図7及び図8に図示したDC値アレイ又は定数オフセットをエッジ・テーブルの代わりに使用するようにもできる。

【0048】図7は、仮想画像70で示した画像を含む圧縮データファイル76から仮想画像70の一部71を抽出するためにポインタ・アレイ・テーブル72とDC値アレイ・テーブル74を用いることを示している。部分71の右下のブロックなどのブロックを圧縮データファイル76から抽出する際には、仮想画像70内のブロックの位置を用いてポインタ・アレイ・テーブル72内にインデックスを作成し、そのブロックを含むMCUへのポインタを返し、また、DC値アレイ・テーブル74内にインデックスを作成してそのブロックのDC値を返す。返されたポインタを用いて圧縮データファイル76内のブロックの圧縮MCUを特定し、返されたDC値をそのブロックの右側のブロックのDCオフセットとして、JPEG規格に準拠するように用いる。つまり、取り出したブロックを展開する際に、逆コサイン変換のDC値は返されたDC値と圧縮MCUとともに保存してある差分DC値を加算することで求まる。これ以外で、1段階の節約には、DC値アレイが各々のブロックについての絶対DC値を含め、逆コサイン変換用のDC値をアレイからの値とブロックからの値を組み合わせる代わりに直接DC値アレイ・テーブル74から読み出せるようにする。

【0049】図8は、DC値アレイ・テーブル74に代えて、固定DCオフセットの使用を図示したものである。オフセット78は圧縮画像内のブロックから読み出した差分DC値の全部に加算する単一の値を提供し、一方、DC値アレイ・テーブル74は各々のブロックについて独自の値を提供する。オフセット78を用いる利点は、DC値に必要な記憶容量を単一の値に減少することである。場合によってはオフセット値0が好適であり、このような場合にはオフセット78とこれを加算する段階をどちらも排除することができる。

【0050】完全なDC値アレイ用にさらなる記憶空間を割り当てても、圧縮画像は画像全体の占有空間より少ない空間に納まる。前述の8.5インチ×11インチの大きさで400画素/インチの解像度、2:1:1サブサンプリング、1バイト/成分の画像を例にとると、圧縮していない画像は、93.5平方インチ×160,000画素/平方インチ×3成分/画素×1バイト/成分、又は、44.88メガビットを占有することになる。64画素/ブロックの場合、画像は467,500ブロックを必要とする。各々のMCUが2ブロックで4バイトのポインタ1つと2バイトのDC値2つ（MCUの各ブロックに1つずつ）を必要とする場合、ポインタ・アレイ・テーブル72は各々が4バイトのエントリ233,750個を有するので935,000バイトを必要とし、DC値アレイ・テーブル74は各々2バイトのエントリ467,500個を有するので935,000バイトを必要とする。全体の圧縮比が10:1の場合、ファイル76に保存された圧縮画像（4.488メガバイト）と、ポインタ・アレイ・テーブル72と、DC値アレイ・テーブル74との占有空間は6.358メガバイトでしかなく、さらに圧縮比は44.88:6.358又は約7:1である。

【0051】図9は、図10に図示した編集システム100が実行できる編集処理のフローチャートである。処理はステップ80で始まり、ステップ98へ進むように図示してある。ステップ80から、編集システム100はステップ82へ進む。

【0052】ステップ82では、編集システム100は編集関数を選択してステップ84へ進む。典型的な編集関数には、パン（移動）、伸縮、回転、マスキング（隠蔽）が含まれる。幾つかの実施例ではこの編集関数は予め設定してあり、また、別の実施例では、通常入力画像又は出力画像の一部のどちらかを観察してから編集関数を利用者が選択する。

【0053】ステップ84では、編集システム100は処理しようとする第1の出力ブロックを指し示すように出力ブロック・ポインタを初期化し、ステップ86へ進む。

【0054】ステップ86では、編集システム100はどの入力ブロックを処理して現在の出力ブロックを生成

するかを決定し、ステップ88へ進む。入力ブロックは現在の出力ブロック・ポインタと特定の編集関数をもとに決定される。例えば、編集関数が色の変更だけの場合には、出力ブロックは編集関数と単一の入力ブロックの関数であるが、編集関数が変換の場合には、出力ブロックは4つまでの入力ブロックの関数となることがある。

【0055】ステップ88では、編集システム100は出力ブロックを生成するのに必要な入力ブロックを抽出展開し、ステップ90へ進む。抽出すべき入力ブロックはポインタ・アレイ内でブロックを含むMCUについてポインタを特定しそのポインタを用いて圧縮画像内のMCUをアドレスすることによって発見可能である。入力ブロックの展開処理において、その入力ブロックについてのDC値（絶対値又は相対オフセットの何れか）はDC値アレイから読み出される。このステップでは、キャッシュ114を編集システム100に設けることによって使用する展開入力ブロックを同じ入力ブロックに依存する他の出力ブロックとともに記憶しておくこともできる。

【0056】ステップ90では、編集システム100は入力ブロックと編集関数に基づいて出力ブロックの画素値を計算し、ステップ92に進む。ステップ92では出力ブロックを出力して出力画像ファイルを形成する。

【0057】次に、ステップ94では、出力ブロック・ポインタをインクリメントして次の出力ブロックを指すようにし、さらに、ステップ96ではさらに計算しなければならない出力ブロックが他にも残っているか否かによって、流れは次の出力ブロックがある場合にはステップ86へ戻り、又は、ステップ98で処理を終了する。

【0058】図10は、圧縮入力画像ファイル101を圧縮出力画像ファイル102へ編集するための本発明による編集システム100を示す。幾つかの実施例において入力と出力の画像は、入力MCUの集合として一緒に保存されており、出力画像は入力MCUを置き換えるか、又は、図1の圧縮画像メモリ12で図示したように入力MCUに追加するかの何れかとなる。

【0059】さらに図10を参照すると、編集システム100は入力ブロック選択部106と、ブロック展開部108と、ブロック単位の画像処理部110と、レジスタ112と、任意のキャッシュ114と、メモリ118とを含む。アレイ生成部120を設けて入力ブロックへのポインタを含むポインタ・アレイ・テーブル72を格納しメモリ118にはDC値アレイ又は固定DCオフセット値のどちらかを格納する。利用者I/O130も設けて編集システム100の利用者が編集関数を選択し入力及び／又は出力画像ブロックを観察できるようにする。

【0060】入力ブロック選択部106は、編集関数の指示が利用者I/O130又は別の供給源から供給される線104からの入力と、現在処理中の出力ブロックに

についての指示を提供するレジスタ112からの入力を受け取る。入力ブロック選択部106はブロック選択線122上にどの入力ブロックを選択したかを表わすための出力を有する。ブロック選択線122はポインタ・アレイ・テーブル72をアドレスするために用い、DC値アレイを用いる場合にはメモリ118をアドレスする。入力ブロック選択部106も、圧縮入力画像ファイル101から選択したブロック/MCUを受け取るための入力と、線124上でメモリ118からのDCオフセットを受け取るための入力と、選択したブロック/MCUをブロック展開部108へ提供するためにこのブロック展開部108へ結合してある出力とを有する。別の実施例において、線124はブロック展開部108へ直接結合し、入力ブロック選択部106はDC値を計算しない。

【0061】ブロック展開部108は、展開したブロックを画像処理部110へ出力するための出力を有し、幾つかの実施例においては、展開したブロックを出力するためのキャッシュ114への出力とキャッシュ114から展開したブロックを呼び出すための入力とを含む。キャッシュ114は回路の寸法より処理速度が優先するような場合と、編集関数が入力ブロックを反復的に使用するとわかっている場合に用いる。

【0062】画像処理部110は、線104から選択した編集関数の指示を受け取るための入力と、出力ブロックを出力するための出力とを含む。レジスタ112の内容は圧縮出力画像ファイル102へ提供されて、画像処理部110の出力するブロックが正しい位置に保存されるが、幾つかの実施例では、内容を画像処理部110へ提供し、ここでさらに圧縮出力画像ファイル102へ出力するブロックへアドレスの指示を添付するようになっている。さらに別の実施例においては、出力ブロックのアドレスはブロックが出力される順序で示される。

【0063】アレイ生成部120は、圧縮入力画像ファイル101から情報を受け取るための入力と、ポインタの値をポインタ・アレイ・テーブル72へ出力するための出力とメモリ118へDC値を出力するための出力の2つの出力を有する。アレイ生成部120を他の部材へ結合する線は、これらの線がポインタ・アレイ・テーブル72とメモリ118の初期化の間にだけ使用され、初期化以後は不要であることを示すために破線で示してある。これは、ポインタ・アレイ・テーブル72とメモリ118が更新されないということではない。出力画像と入力画像が結合されるような編集システムの一部として実装する場合、ブロックを編集して入力画像ファイル内で移動するとポインタ・アレイとメモリ118を更新するような機構（図10には図示していない）を設ける。

【0064】動作において、レジスタ112は注目する第1の出力ブロックへのポインタで初期化され、ポインタを介して注目するその他のブロック全部を走査するようにプログラムする。例えば、仮想画像70の一部71

(図7及び図8参照)を編集関数で操作しようとする場合、レジスタ112は部分71の16個のブロックの各々を順次指し示すことになる。線104上に指示された特定の編集関数とレジスタ112によって示される出力ブロックから、入力ブロック選択部106はどのブロックを入力するか決定する。例えば、選択した編集関数が仮想画像70の中心にある画素を通る反映(鏡像)で現在の出力ブロックが左下隅の画素だとすると、選択すべき入力ブロックは右上隅の画素となる。他の編集関数、例えば、回転、伸縮、及び小数ブロック変換などでは、1つ以上のブロックを現在の出力ブロックとして選択することがある。

【0065】入力ブロックを選択した後、入力画像ファイル101からこれらを抽出する(入力ブロック選択部106がキャッシュ114にも結合しており、選択したブロックが既にそこに記憶されている場合を除く)。ブロックを見つけ出すには、入力ブロック選択部106がそのブロックを含むMCUのMCUアドレスをブロック選択線122へ出力し、ポインタ・アレイ72が選択したMCUへのポインタを提供する。選択したMCUは選択したブロックについてのDC値オフセットと併せてブロック展開部108へ供給される。幾つかの実施例において、DC値はブロックをブロック展開部108へ送出する前に圧縮入力画像ファイル101内でブロックとともに保存してあるDC差分値と結合する。

【0066】選択したブロックをブロック展開部108で展開した後、展開したブロックは画像処理部110へ供給される。ここで、従来技術に対して編集システム100が提供する1つの改良は、画像全体を処理しなければならない場合であっても、圧縮入力画像ファイル101全体を一度に全部展開する必要がないことである。これは、通常であれば、画像処理部110の何らかの所定の演算には画像の小さな部分集合だけしか必要とされないためである。画像処理部110は出力ブロックを出力するが、これは編集システム100の動作により、選択した入力ブロックと選択した編集関数だけからなる関数である。編集システム100が提供するさらに別の改良は、メモリ118を正しく初期化すれば、選択したMCUは選択したMCU以外のどのMCUも参照することなく圧縮入力画像ファイル101から抽出されることである。エッジ・テーブル又はDC値アレイを用いない場合には、選択したブロックより先のブロックを取り込んで選択したブロックの正しいDC値オフセットを求めなければならないことが多い。

【0067】図11は、DC差分値計算回路140のブロック図で、レジスタ142、144と、2対1マルチプレクサ146と、差分加算回路148とを含む。レジスタ142は1周期の遅延を有しているもので、End\_Of\_Row信号がレジスタ142のCLR入力に発行され値0を保持する場合を除き、直前に適用したDC値を

保持する。入力線150はブロックのDC値をレジスタ142の入力と加算回路148の加算入力へ提供するために使用する。レジスタ142、144の出力は各々マルチプレクサ146の入力へ結合しており、マルチプレクサ146の出力は加算回路148の減算入力へ供給する。マルチプレクサ146はJPEG互換モードと固定DC値モードの一方を示すモード選択入力を含む。

【0068】このようなDC差分値計算回路140は、圧縮しようとするブロックとともに保存するDC差分値を計算するために使用する。マルチプレクサ146がJPEG互換モードに設定してある場合、レジスタ142からの値が加算回路148の減算入力へ送られる。即ち、加算回路148から出力されるDC差分値は直前のブロックのDC値を引いたブロックのDC値、又は、そのブロックがある行の第1のブロックの場合には0である。マルチプレクサ146が固定DC値モードに設定してある場合は、レジスタ144からの値は加算回路148の減算入力へ送られる。この場合、各々のブロックのDC値はレジスタ144に保存してある定数値を減算してから出力される。定数値は、例えば、画像内のブロックの絶対DC値全部の平均値とするなど幾つかの方法で選択できる。選択した定数が圧縮画像内部に含まれるデータの単なる関数ではない場合には、画像に追加のフィールドを付与してアレイ生成部120による後の抽出のための定数を保存しておき、アレイ生成部120が値をメモリ118へ抽出できるようにする。

【0069】上記の説明は、解説のためであり、これに制限するものではない。本発明の多くの変形が、本開示の参照により、当業者には明らかとなろう。単なる例として、画像処理部14の別個に識別されたユニットがコンピュータ上で動作する1つ又はそれ以上のソフトウェアプログラムとして実現することができ、又は、このユニットの機能を統合又はさらに細分することが有り得る。従って、本発明の範囲は、上述の実施例上の説明を基準とすることなく定めるべきものであって、特許請求の範囲の記載を参照し、これと等価の範囲において決定されるべきものである。

【0070】

【発明の効果】本発明によれば、上述したように構成したので、圧縮された形状のままの画像の画素レベルでの操作が可能となり、特に、典型的な画像圧縮方式を使用する画像圧縮システムにおいては、ポインタで圧縮画像ファイル中のMCU(最小符号化単位)を示すポインタ・アレイを用いており、幾つかの発明においては、DC値テーブルを設けてあるブロックの絶対DC値が他の何らかのブロックにあるデータを参照する必要なしに計算することができる。つまり、画像のブロック全体からブロックの部分集合を仮想画像として選択することができる。幾つかの発明において、仮想画像は1つの仮想画像より大きい画像領域を包括するように移動し、場合によ

って、画像領域全体を順次包括することもできる。

【0071】また、本発明によれば、仮想画像を展開／復号して編集し、編集した各々をMCUに圧縮し、編集したMCUを指すように本来のMCUへのポインタを変更して、編集した画像データ領域に保存するので、ポインタ・アレイは取消操作を実行するように変更することが可能となり、対話型画像編集アプリケーションにおいて有用な方法となる。当然のことながら、対話型画像編集アプリケーションに代えて、独立した出力画像ファイルを作成する逐次処理によって入力画像ファイル全体を編集することも可能である。

【0072】さらに、本発明によれば、圧縮されたMCUから全てのMCUが復号されないようなMCU内の差分符号化した値を取り扱うための手段も提供されている。例えば、値を保持するためにエッジ・テーブルを用意し、ここで各々の値は仮想画像の辺縁上のブロックについての差分値と結合された場合、仮想画像の辺縁より遠方のブロックを参照することなくそのブロックについての絶対値を提供することができる。エッジ・テーブル内のエントリは、ブロックを完全に展開せずに圧縮されたままのMCUから決定できる。これは、逆コサイン変換がエッジ・テーブルの作成に必要なためである。さらなる記憶空間の必要性より処理速度が重要な場合に差分値から絶対値を計算する速度を向上させるため、1つ以上のエッジ・テーブルを設けることもできる。絶対値計算は平均してエッジ・テーブルが多いほど高速であるが、これは、計算速度がエッジ・テーブルを有する最も近い辺縁までのブロックからの距離に比例するためである。

【0073】また、本発明によれば、処理速度と記憶空間が両方とも優先する場合、差分DC値は一定のDC値に対する差分としているので、JPEG互換ではないが、各々のMCUが独立して展開自在であり、エッジ・テーブル又はDC値アレイを不要にすることもできる。この場合に用いる画像ファイルとJPEG互換画像ファイルの間の画像の変換は、計算が簡単で、無損失、反転自在な演算となる。

【0074】また、本発明によれば、全てのMCUを並べ替え原本の画像データ領域からのMCUを編集した画像データ領域からのMCUで置換するために圧縮された編集画像の再インデックスを取るための手段も提供されている。例えば、JPEG規格に従って圧縮された画像データで動作するが、本発明によれば、ブロックが可変容量MCUに圧縮されるような、又は、データが差分値で符号化されることによりMCUを完全に復号するためには別のMCUを参照しなければならないようなあらゆる画像圧縮方式についての応用が可能となる。

【0075】一方、本発明による仮想編集装置によれば、圧縮画像は主メモリ領域内に読み込まれ、また、編集するために選択したブロックが主メモリ領域から展開

されてエディタに渡され、編集され、コンプレッサ回路によって再圧縮された後、このように再圧縮され、編集されたブロックは編集ブロックメモリ内に配置されるので、ポインタ・アレイは、編集されたブロックメモリ又は主メモリ領域内の何れかのブロックを示し、アレイ内のポインタが編集後の画像の各々のブロックを指し示すために提供されるものとなる。

#### 【図面の簡単な説明】

【図1】本発明の一実施例を示す仮想編集システムのブロック図である。

【図2】ディスプレイ・メモリのメモリ・マップと2次元アレイ中の圧縮画像を示す模式図である。

【図3】各々が線形アレイ内のMCUとポインタ・アレイを含む画像ファイルを示す模式図である。

【図4】仮想画像と対応するエッジ・テーブルを示し、エッジ・テーブルの使用を図示した模式図である。

【図5】ポインタ・アレイからのポインタを示す模式図である。

【図6】編集したMCUの編集ヘッダのフィールドを示す模式図である。

【図7】DC値アレイとポインタ・アレイを用いて圧縮画像データファイルから仮想画像の一部を抽出する方法を示し、矢印が各々の関連エントリを示す模式図である。

【図8】DC値アレイとポインタ・アレイを用いて圧縮画像データファイルから仮想画像の一部を抽出する方法を示し、矢印が各々の関連エントリを示す模式図である。

【図9】編集処理を示すフローチャートである。

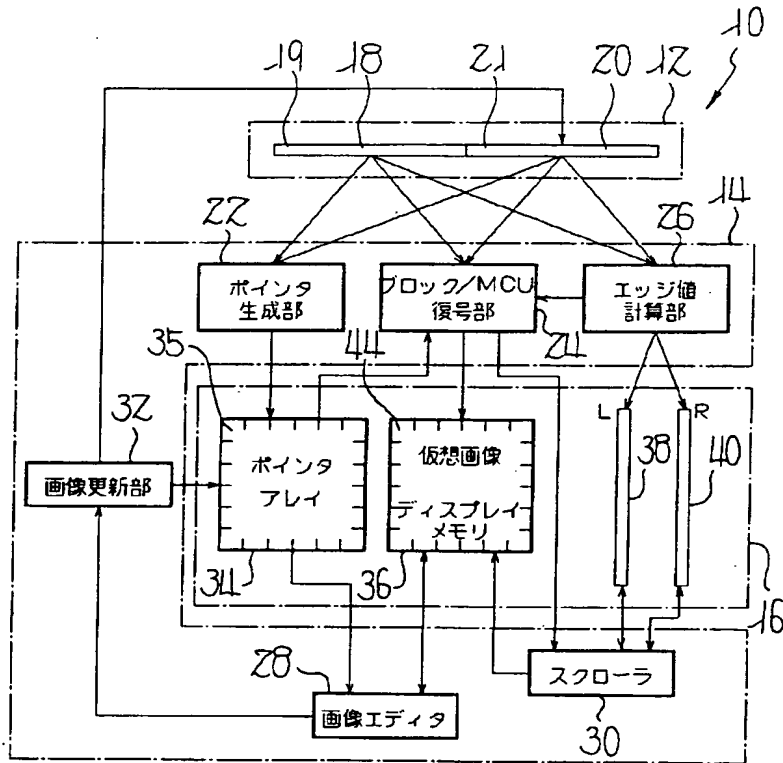
【図10】本発明の他の一実施例の画像編集装置を示すブロック図である。

【図11】圧縮画像に組み込もうとするブロックについてDC値を符号化するために用いる符号化回路を示すブロック図である。

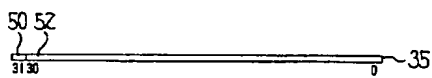
#### 【符号の説明】

12	圧縮データメモリ
16	仮想画像メモリ、編集データメモリ
28	エディタ
34	ポインタ・アレイ
38	左側エッジ・テーブル
40	右側エッジ・テーブル
44	仮想画像
70	仮想画像
72	ポインタ・アレイ
101	入力メモリ
102	出力メモリ
106	選択手段
108	展開手段
110	画像編集手段
114	参照テーブル

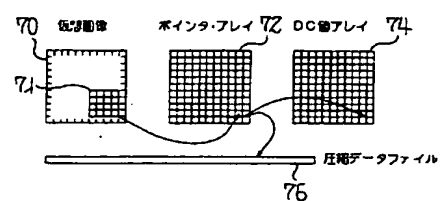
【図1】



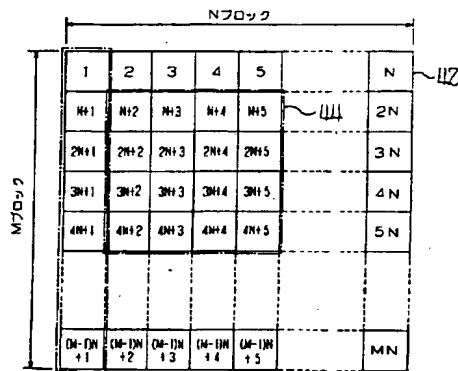
【図5】



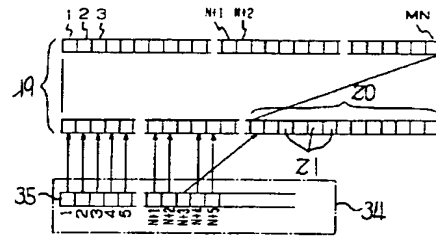
【図7】



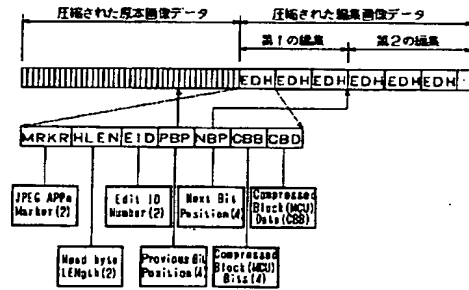
【図2】



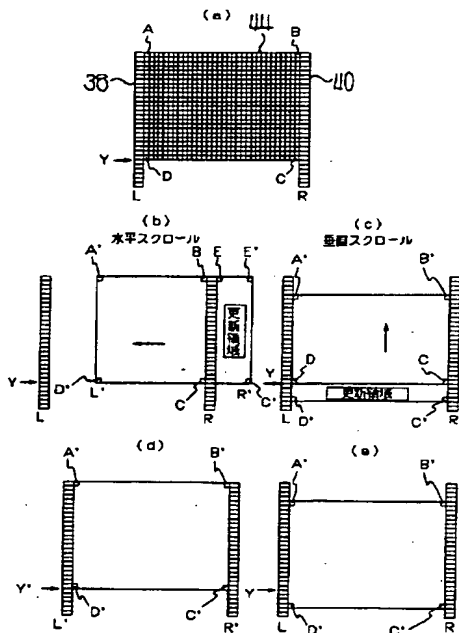
【図3】



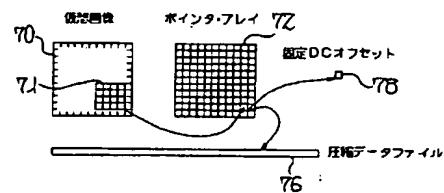
【図6】



【図4】

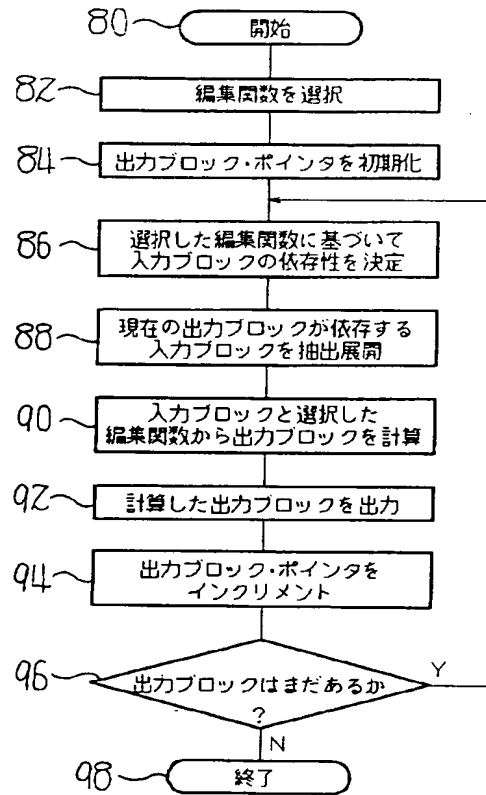


【図8】



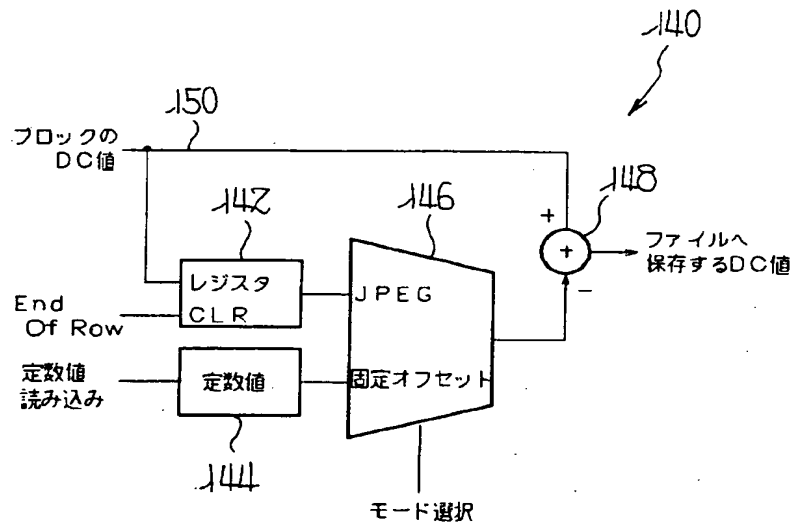


【図9】





【図11】



フロントページの続き

(51)Int.Cl.<sup>6</sup>

H04N 1/41

7/24

識別記号

庁内整理番号

B

FI

技術表示箇所

H04N 7/13

Z

(72)発明者 マーティン ピー ボーリック  
 アメリカ合衆国 カリフォルニア州 メン  
 ロー パークスイート 115 サンド ヒ  
 ル ロード 2882 リコー コーポレーシ  
 ョン カリフォルニア リサーチ センタ  
 内

(72)発明者 スティーブン エム ブロンスタイン  
 アメリカ合衆国 カリフォルニア州 メン  
 ロー パークスイート 115 サンド ヒ  
 ル ロード 2882 リコー コーポレーシ  
 ョン カリフォルニア リサーチ センタ  
 内